Computers and Mathematics with Applications 59 (2010) 2541–2551

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa

Bayesian network modeling for evolutionary genetic structures

Lisa Jing Yan *, Nick Cercone

Department of Computer Science and Engineering, York University, Toronto, ON, Canada M3J 1P3

ARTICLE INFO

Article history:
Received 23 July 2009
Accepted 30 December 2009

Keywords:
Bayesian network modeling
Evolutionary computing
Genetic algorithm
Structure learning
Constraint based
Score based

ABSTRACT

Evolutionary theory states that stronger genetic characteristics reflect the organism's ability to adapt to its environment and to survive the harsh competition faced by every species. Evolution normally takes millions of generations to assess and measure changes in heredity. Determining the connections, which constrain genotypes and lead superior ones to survive is an interesting problem. In order to accelerate this process,we develop an artificial genetic dataset, based on an artificial life (AL) environment genetic expression (ALGAE). ALGAE can provide a useful and unique set of meaningful data, which can not only describe the characteristics of genetic data, but also simplify its complexity for later analysis.

To explore the hidden dependencies among the variables, Bayesian Networks (BNs) are used to analyze genotype data derived from simulated evolutionary processes and provide a graphical model to describe various connections among genes. There are a number of models available for data analysis such as artificial neural networks, decision trees, factor analysis, BNs, and so on. Yet BNs have distinct advantages as analytical methods which can discern hidden relationships among variables. Two main approaches, constraint based and score based, have been used to learn the BN structure. However, both suit either sparse structures or dense structures. Firstly, we introduce a hybrid algorithm, called "the E-algorithm", to complement the benefits and limitations in both approaches for BN structure learning. Testing E-algorithm against a standardized benchmark dataset ALARM, suggests valid and accurate results. BAyesian Network ANAlysis (BANANA) is then developed which incorporates the E-algorithm to analyze the genetic data from ALGAE. The resulting BN topological structure with conditional probabilistic distributions reveals the principles of how survivors adapt during evolution producing an optimal genetic profile for evolutionary fitness.

1. Introduction

Bayesian Network (BN) modeling for evolutionary genetic structure, uses BN to analyze genotype data derived from evolutionary processes and provides a graphical model to describe hidden dependencies among genes. According to evolutionary theory, stronger genetic characteristics reflect the organism's ability to adapt to its environment and to survive the harsh competition faced by every species [1–3]. Each individual's traits and characteristics are coded into cellular information called genes. Genes evolve to be strong, fit genes; that is, nature selects the best genes and reproduces them using inheritance through generations of survivors. Such evolution normally takes millions of generations. But what are the hidden connections which constrain genotypes, yet lead to superior characteristics which promote survival is rather interesting. In order to explore this problem, we accelerate this process significantly, so that we can evaluate the genetic change much more rapidly. We then analyze the hidden evolutionary relationships. Having revealed these connections, we can determine which precise factors and connections promote fitness in an individual population or species.

* Corresponding author. Tel.: +1 4169975002.
E-mail addresses: jingyan@cse.yorku.ca (L.J. Yan), ncercone@yorku.ca (N. Cercone).

0898-1221/$ – see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2009.12.039

There are a number of models available for data analysis such as artificial neural networks, decision trees, factor analysis, BNs, and so on. Yet BNs have distinct advantages as computational tools. BN is an analytical tool which can discern hidden relationships among variables [4]. BN can handle incomplete datasets just as well as complete ones, and it can discover dependencies among all variables by representing them in a comprehensible graphical model.

BNs have been widely used in bioinformatics (gene regulatory networks, protein structure), medicine, document classification, information retrieval and image processing [5–10,24–26]. As probabilistic models, BNs have been used to replace traditional variation of genetic and evolutionary algorithm in evolutionary computing [11]. In [11], Pelikan segments chromosomes to different traps as variables and build a probabilistic model based on this; after that, only use this model to sample the solutions and generate new candidates population. BN has provided a more promising solution population, however, the real reason why this method can bring out the optimal candidates population more efficiently is the discovery of the hidden relationship among the genes. Thus, our work is undertaken as a response to reveal the discovery of this hidden relationship among the genes by applying BN as an analytical tool for a population solution space, rather than a probabilistic sampling tool.

We therefore propose to apply BNs to analyze data arising in genetic research. We demonstrate our idea on a simulated genetic dataset, which mimics a biology-driven artificial life (AL) environment [12]. This AL simulation, Artificial Life Genetic Algorithm Expression (ALGAE), provides a useful and unique set of meaningful data, which can not only describe the characteristics of genetic data, but also simplify its complexity for our BN analysis. BAyesian Network ANAlysis (BANANA) is then developed to analyze the genetic data from ALGAE. BANANA incorporates a BN structure learning algorithm: the E-algorithm, first proposed by Yan et al. [13] and has been proven to be an efficient and accurate algorithm for constructing BN structure by later adaptations, applied to a business model [10,14].

The goal of our research is to reveal the hidden connections among genetic characteristics. Each chromosome in the AL species contains a coded gene sequence representing particular species characteristics. These characteristics appear random, but after generations of evolution, certain genetic attributes will emerge as dominant. However, this hidden information is not apparent from the raw data, and the meaning needs to be extracted and interpreted. BN analysis of the genetic data can produce a graphical and statistical representation showing the dependencies between genotypes among populations.

The significance of the analysis of the hidden dependencies between genetic descriptors is that two important outcomes are produced as a result of research. Firstly, we generate an interesting and unique genetic dataset using the AL model, which extends the versatility and utility of the Genetic Algorithm (GA) so that it becomes a remarkable instrument for creating hypotheses for any given entities. Secondly, using BN to analyze the hidden dependencies among AL genetic data is a unique methodology. It provides a new approach for problem solving by combining evolutionary principles and BN modeling, based upon generating unique and expressive data.

This paper is organized as follows: Section 2 provides background regarding Bayesian network learning and the E-algorithm; Section 3 introduces the design of ALGAE, and experiments to obtain artificial genetic data; Section 4 explains the process called BANANA, and the modeling for AL genetic data structure, and discusses the experimental results of genotype characteristic hidden connections; Section 5 summarizes our contribution and provides some open questions for further research.

2. Bayesian network learning

Bayesian networks are a graphical representation of probabilistic causal relationships among random variables (factors). A BN has two components: a topological structure and its conditional probability distribution (CPD). The BN structure is an acyclic directed graph in which each vertex i corresponds to a random variable X_i. An arc $X_i \rightarrow X_j$ describes the dependency between variable i and j. This dependency also states the causal relationship between them, thus, variable i is the parent node of j, and variable j is the descendant node of i. In this graph, each vertex i is attached with its conditional probabilistic distribution $p(X_i|\Pi_i)$ of X_i given its parents Π_i. We assume that each variable is probabilistic independent of its non-descendants given its parent states. Thus, the joint probability distribution $\mathcal{P}(X)$ for all the variables X [15], can be described as follows in Eq. (1):

$$\mathcal{P}(X) = \prod_{i=1}^{n} p(X_i|\Pi_i). \tag{1}$$

The advantage of a BN is that it can describe data in both qualitative and quantitative aspects. Qualitatively, a BN structure gives data a graphical interpretation which can be understood easily; and quantitatively, CPD describes strength of the causal relationships among the factors. Thus, learning Bayesian networks can be examined as the combination of parameter learning and structure learning. Parameter learning is to estimate the conditional probabilities (dependencies) in the network, whereas, structural learning is to estimate the topology (arcs) of the network. This following section discusses how to learn Bayesian network structures from data.

2.1. Basic approaches for BN structure learning

Given a set of variables and a dataset composed of all these variables' values, the problem is to build a structure to present the connections among the variables. This structure learning process needs to select the arcs between them and estimate

the parameters. Developing a structure is very useful for a variety of applications in general, for example, where there are masses of data available and we want to understand what underlies the knowledge or what attributes are correlated. In addition to providing a model that will allow us to predict behavior under conditions that we have not seen, the structure can also incorporate domain expert knowledge to provide more reliable suggestions. However, to include all the information from the data into the structure, yet to keep the structure simple and condensed with only critical information, is going to be a trade-off problem. Two main approaches are used to learn structure in BNs: the constraint-based and the score-based approaches.

a. Constraint-based approach:
The constraint-based approach poses learning as a constraint satisfaction problem, which is more intuitive and follows the definition of a BN more closely. This method performs tests of conditional independence (CI) on the data, and search of for a network that is consistent with the observed dependencies and independencies [16,15,5].
As a typical metric, CI is based on the metric of information flow in information theory [17,18], thus the mutual information of two variables X, Y is defined as Eq. (2):

$$I(X, Y) = \sum_{x,y} P(x, y) \log \frac{P(x, y)}{P(x)P(y)} = \sum P(x, y)I(x, y) \tag{2}$$

and conditional mutual information is defined as Eq. (3):

$$I(X, Y|C) = \sum_{x,y,c} P(x, y) \log \frac{P(x, y|c)}{P(x|c)P(y|c)} = \sum P(x, y)I(x, y|c) \tag{3}$$

where C is a conditional set of nodes, P denotes the instance frequency (probability) observed from a sample dataset. The mutual information can show if the two variables are dependent and if so, how close is their relationship. Hence, when $I(X, Y|C)$ is smaller than a certain threshold value ε, we can say that X is independent of Y given the set C, or else X is dependent of Y if C is an empty node. So we can deduce if there is a connection between two variables in view of the mutual information.
Here, the threshold value ε can be given based on expert knowledge, alternatively, there is another similar method, the χ^2 test [19], which is based on a statistical hypothesis to estimate a connection between two variables. Given a degree of confidence σ, a connection between two variables can be deduced by t-value (threshold) which is generated by χ^2 test. In our case, if the connection value I is greater than or equal to t-value, then X is independent of Y, which implies that there is no direct connection between these two variables. Otherwise, if the connection value I is less than t-value, then X is dependent of Y, which means that an arc connects X and Y in the resultant network.

b. Score-based approach:
The score-based method is to define a score function that evaluates how well the dependencies in a structure match the data, and search for the simplest structure which also maximizes the score. In the set of feasible solutions, a recursive search can be used to find an optimal structure that satisfies the criteria. A scoring function commonly used to learn BN is the log-likelihood, which is simply the log of the likelihood function, that is, Eq. (4):

$$l(X|g, \theta_g) = \log \prod_{i=1}^{n} p(X_i|\Pi_i, g, \theta_g) \tag{4}$$

$$= \sum_{i=1}^{n} \log p(X_i|\Pi_i, g, \theta_g), \tag{5}$$

where, θ_g is a parameter of the structure g in a dataset X which also represents all the variables. The log-likelihood is easier to analyze than the likelihood, because the logarithm turns all the products into sums. Therefore, according to Eq. (4), we have Eq. (5).
There are a couple of important points to note about the log-likelihood. The log-likelihood increases linearly with the size of data. The higher scoring networks are those where the node and the parents are highly correlated. The network structure that maximizes the likelihood is often the fully connected network. Adding a node into the networks always increases the log-likelihood. This deficiency of the log-likelihood score is not desired. Thus, a score that makes it harder to add arcs is necessary. In other words, we would like to penalize structures with too many arcs. One possible formulation of this idea is called the minimum description length (MDL) score [20]. The MDL score is a compromise between fit to data and model complexity. Adding a variable as a parent causes the log-likelihood term to increase, but so does the penalty. There will be an arc addition if its increase to the likelihood is worth it. The detailed MDL scoring function will be explained in the following section.

The space of Bayesian networks is a combinatorial space, consisting of an exceeding large number of structures. This problem is combinatorially complex; both approaches have their limitations.

The constraint-based method requires that conditional independence relationships between attributes first be determined, and then a structure which satisfies them is developed. This approach is problematic since conditional

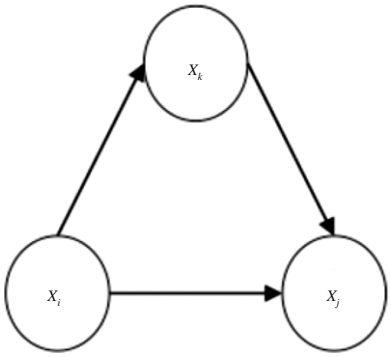


Fig. 1. Δ-form.

independence relations are difficult to achieve with certainty. When it comes to a sparse structure, the constraint-based approach could be efficient. Otherwise, not only will some dependency test results be inaccurate, but also an exponential number of dependency tests have to be performed.

Scoring methods use score functions which can determine structures through a metric, and the advantage is that they are less sensitive to errors in individual tests. In general, the problem of finding the best-scoring network structure is NP-hard. Some heuristic information can be used and to reduce the search space of BN structures [5,16,19,20,13]. However, for the score-based approach, the cost of computation is too high in a huge search space when the conditional set is large, which proves problematic [20,10,13].

As each approach has its own disadvantages, many hybrid algorithms uniting these two approaches have been developed in the last decade [8,19,21]. The general idea is quite straightforward. First, the constraint-based tests are performed to get an initial network to consider, which reduces the search space. Then, a metric score function is used to find a matching structure which has the best motivated score.

2.2. E-algorithm

The key aspect of the structure learning problem is to construct a topology network from fully observable variables. This section provides an improved BN learning algorithm: the E-algorithm, which firstly proposed in [13] undertaken in relation to improving learning Bayesian networks. The E-algorithm has been adapted to business applications, e.g., suggested business strategies that a business should choose, as reported in [10,14]. In [23], the accuracy and efficiency of the E-algorithm has been established by comparing execution time of the E-algorithm against two established algorithms: I-MDL, I-B&B-MDL.

The following section introduces the E-algorithm. The E-algorithm has two main contributions:

(1) The constraint-based algorithm, by using a set of lower order independence tests (χ^2 test), restricts search space and enhances search efficiency. It computes the mutual information among variables to construct the initial network, and limits the possible parents of each node. Note that the overall variables X are in a sequence; any node X_j ($j > i$) appears after X_i will not be X_i's parent node. Thus, instead of having $i-1$ potential parents for node X_i, the algorithm only considers k ($k \ll (i-1)$) possible parents in each search. Since the search space is significantly restricted, the search is more efficient. Thus, the E-algorithm defines a new "Δ-form" structure in a BN and brings it in to restrict the search space. The definition of "Δ-form" is as follows:

Given an arc between two nodes X_i and X_j in BN structure g, if there is another path connecting them which only includes one extra node X_k, we call this acyclic subgraph an order-1 "Δ-form" (Fig. 1); if this path includes two extra nodes, we call this subgraph order-2 "Δ-form".

(2) Although the Branch&Bound-MDL-based learning algorithm improved the search aspect of the MDL-based learning algorithm, two problems still exists for Independent-Branch&Bound-MDL(I-B&B-MDL) when the number of nodes is large. One problem is that the number of conditional sets may be too large, resulting in extra time needed to collect data and compute mutual information, even if only performing lower order independence tests. The second is that, as there is an extra cost in CI tests, the algorithm cannot ensure that there are enough pruned sub-trees to make I-B&B-MDL more efficient than ordinary B&B-MDL. Thus, combining MDL metric scoring closely with CI test in the local "Δ-form", the E-algorithm brings specified conditional mutual information test, and determines each node's parents' ordering as heuristic information to reduce recursive search in the search space, in order to find a fit structure for the given dataset effectively.

We will introduce the design and implementation of the E-algorithm, and also demonstrate its validity and reliability for recommending gene expressions.

2.2.1. Description

Combining both a constraint-based approach and also a score-based approach, the E-algorithm jointly applies the CI test and MDL metric search. First, a small number of dependence tests are used to reduce the calculation complexity and to

restrict the feasible search space. Second, the improved MDL metric search boosts both time performance and efficiency of BN learning.

The E-algorithm considers the BN structure learning as a connection elimination process starting from a fully connected graph G_0 among all the variables. It features three elements: (1) order-0 independence tests are used to delete week connections and obtains a graph G_1; (2) order-1 and order-2 conditional independence χ^2 tests, which only appears in the "Δ-form", are conducted and simplify G_1 to G_2, which reduces the search space for scoring possible structures. (3) by means of ordering mutual information, the sort order for candidate parent nodes increases the cut-offs of B&B search tree and decreases the number of redundant recursions in order to accelerate the search process. The E-algorithm then directly evaluates the structure MDL scores by using parents' ordering as heuristic information, to accelerate the search process without redundant recursions. Eq. (6) defines a score that evaluates how well the dependencies in a structure match the data, and search for a structure that maximizes the score [19,20].

$$MDL(g, X) = \sum_{i=1}^{n} H(i, g, X) + \frac{k(g)}{2} \log n, \tag{6}$$

where $MDL(g, X)$ is the description length of graph g for overall data variables X, $H(i, g, X)$ describes the empirical entropy of each node i and its *sum* stands for the overall structure fitness to the observed data, and $k(g)$ is the description for the complexity of nodes (each node i has the number v_i values, j is a parent node of $i, j = [1, i-1]$), as follows:

$$k(g) = \sum_{i=1}^{n} k(i, g) \tag{7}$$

$$k(i, g) = (v_i - 1) \sum_{j=1}^{i-1} v_j. \tag{8}$$

As we see, the problem of learning BN becomes a search problem for a structure with MDL metric. A recursive search is applied to the MDL-based search procedure. This search examines all possible local changes in the set of parent nodes, revealing that the cost of those evaluations is too high for massive datasets.

In order to reduce the computational complexity for empirical entropy, a B&B-MDL-based algorithms [20] is used to prune worthless recursive calls for certain branches on a search tree by estimating the *MDL* score. Specifically, if the value of MDL_1 in the previous step is smaller than the lower bound value of MDL_2 in the current step, then the further recursive calls in this current branch can be ignored. As the structural complexity increases, along with the number of each node's parent nodes, the value of overall empirical entropy H descends monotonically and it is nonnegative. Furthermore, the decrease of empirical entropy H is the current node i's empirical entropy $H(i, g, X)$. Hence, for a new additional parent node, if

$$H(i, g, X) \leq \frac{k(i, g)}{2} \log n$$

which means $k(i, g)$ (the complexity of the node) has increased more than the improvement of structural fitness to the observed data. Thus, this branch, starting from adding this current node as the branch node's parent node, needs to be pruned.

2.2.2. Algorithm procedure

The E-algorithm is summarized as the following steps.

Step 1: For fully observed variables X (known sequence), conduct order-0 CI tests for each pair variables using Eq. (2), and build an initial graph g_0; each arc meets the constraint condition: $I(X_j, X_j) \geq \varepsilon$ (ε is threshold value), and keep an record of each arc's mutual information in G_0.

Step 2: Conduct order-1 CI tests which appears in a "Δ-form", and compute the conditional mutual information in light of Eq. (3), and remove any invalid arc by t-value which is generated by χ^2test according to a given degree of confidence σ; Simplify G_0 to G_1; Repeat for order-2 "Δ-form" and obtain G_2;

Step 3: For each node X_i, ascertains its candidate parents Π_i according to G_2, and sorts its potential parent nodes as ascending ordering by their mutual information; then adopt the B&B-MDL technique to search from top down, find a Π_i with the minimum MDL score and confirm the local optimized structure of X_i.

(See Appendix: experimental results and analysis on ALARM datasets.)

3. Genetic algorithm in artificial life

We exploit the Artificial Life concept by building a simple ecology system: ALGAE (Artificial Life Genetic Algorithm Expression), and use it to provide a useful and unique set of meaningful data, which can not only describe the characteristics of real genetic data, but also simplify the complexity of data in order to expedite our BN analysis. Here are two aspects considered in this section: firstly, we show the environmental factors which determine the living conditions of the two

Table 1
32-bit chromosome descriptor.

Gene	Description	Bit site	Gene	Description	Bit site
SP	SPecies type	0	CA	Action characteristics	13–15
SL	Life span	1–4	CR	Capricious rate	16–18
VF	Vision field	5–6	SA	Attack speed	19–21
TM	Transition movement	7–8	DA	Defend ability	22–24
CM	Motion characteristic	9–11	LA	Attack loss	25–27
LM	Motion loss	12	EF	Food efficiency	28–31

species who are the subject of the experiment; secondly, we explore the key genetic factors for survival, with details about the chromosome and its variability in the evolutionary process. Then we use BN to show relationships between the genetic factors, and our intent is to reveal the hidden dependencies among the variables which emerge during evolution of the species.

3.1. ALGAE

In ALGAE, certain resources must exist, and these resources are distributed in a two-dimensional grid according to certain rules, as detailed below. We stipulated two kinds of species in this virtual world: Species 1 and Species 2. They survive in the virtual environment through competition for resources (food, mates, and territory) and obey certain rules: species mate within their own species only, males with females; each one subsist on native plant materials, and eat the cadavers of the competitive species as a form of nourishment; when energy levels reach zero, an individual dies and becomes a source of food; also, ages increase until they reach the maximum possible life span, then natural death occurs; barriers are also placed in their living space to constrict their movement.

All behaviors above indicate that the two species compete for resources to survive. As the population evolves, the distribution of resources and barriers changes over time. We examine a population of artificial AChromosomes which present each individual G_i in both species, as below: (Table 1)

$$G_i = [SP, SL, VF, TM, CM, LM, CA, CR, SA, DA, LA, EF], \quad i = \{1, 2\}.$$

ALGAE incorporates the genetic algorithm (GA) for moving from one population of chromosomes (binary value of 0 or 1bit strings representing organisms) to a new population, which uses selection together with the genetic operators of one-point crossover, bit-flip mutation, and inversion. The Fitness function selects the most fit individual, whose genes are carried forward in the evolutionary time frame. A fitness value or score is assigned to each solution, representing the abilities of an individual to 'compete'. The individual with the optimal (or near optimal) fitness score is sought. We further define fitness as survivability. Individuals in a population compete for resources and mates, and those who cannot survive are not fit, in the evolutionary sense, so will become extinct. We splice and segment chromosomes to mimic mutation and natural evolution. Iterated over 120 generations, the result is a chromosome comprising the best genes which have evolved to foster survival fitness through the two species evolutionary process.

In ALGAE, we consider the following aspects, such as living environment (or lifespace), resources, barriers, competition, behavior patterns and preferences, and physical status. The details will be discussed below.

a. Artificial Environment (*AEnvironment*) is defined as a search space designed in a two-dimensional field, a rectangular region with two-dimensional vectors, for directional movement toward a desired object.
b. Assume *resources* exist in the AEnvironment composed of $n \times m$ grids (here we use 32×60), randomly distributed and which are renewable. Two types of food are available to increase energy: plant food (available in certain areas), and animal food (specifically the cadavers of dead competitors).
c. Physical *barriers* exist in an AEnvironment, randomly placed according to rules. These obstacles hinder individual motion; then individuals need to go around the obstacle to gain food, or to copulate or attack. (Note that the number and area of barriers must be less than 50% of the lifespace of the species.)
d. *Competition* is also intrinsic in an AEnvironment. Individuals attack the other species based on Attack Speed (SA) and Defense Ability (DA). They have a certain amount of energy which is lost by movement (Motion Loss, LM), and attack (Attack Loss, LA). Species also gain energy by consumption of food (Food Efficiency, EF). Food is assigned simulating natural law with corresponding food value and vitality. Thus food energy values expire at a particular time limit: the vegetative food time limit (TL1) is 20, animal food time limit (TL2) is 5. Fresh food will increase along with the generation increase, and surpass its limited food supply. Each individual is a gene disseminator, an intelligent individual, facing a complex environment, so choosing suitable adaptive behavior is very important. Appropriate behavior ensures genetic replication and thus evolution. To achieve survival and multiplication, the species member undertakes migration, looks for food, exhibits breeding behavior. Also, in order to ensure the population's evolution, ALGAE programs in mutual attacking behavior which can eliminate the genetically inferior individual.

e. Individual behavior patterns and preferences are programmed as movement modes and action modes into their genes, as follows: (1) Species can only mate with local individuals within their action field. Each individual complies with its own motion characteristic (CM) to choose behaviors: look for food, attack/defend, or mate. In the hypothetical AL world, motion characteristic emulates biological drives. (2) Transition motion (TM) choice, according to the GA aspect of ALGAE, determines that transition motions are all caused by corresponding instinctive (genetically determined) decisions. (3) the action characteristic (CA) gene mimics biological behavior priorities. (4) Capricious Rate (CR) indicates that decisions made by individuals can be unpredictable and capricious; as in real life, individuals do not have to comply with the normal order of things, given that there are sometimes peculiar circumstances in which our behavioral characteristics allow freedom to choose our own behavior.
f. Physical status such as life span (SL) is also genetically determined. When a certain age is reached, or energy entropy reaches a threshold, the individual dies. Individual age increases along with the generation increase, surpassing the life span, ending in natural death. Regarding the (biological) initial age, in order to simulate the initial population subject to the process of evolution, individual age is assigned as a random number — the biological minimum age (SL_{MIN}). Similarly, the initial biological energy available is stated as $Energy = 70 + random(30)$ (Maximum energy is 100) to ensure a level of individual energy consumption during the initial migration.

3.2. ALGAE run process

In ALGAE, the program establishes the artificial world (AWorld) environment parameters, comparable to biological evolutionary pressure. Using a graphical interface dynamic demonstration, it records the evolutionary processes for each generation (which survives).

The system operation follows basic steps and establishes parameters in relation to the environment as follows:

Step 1. Initialize AWorld environment, randomly set up barriers and vegetative food supply;
Step 2. Initialize a population of AChromosomes randomly, with each individual $i\,Energy_i$ between 70 and 100, and Age_i between 0 and SL_{MIN};
Step 3. Evolutionary process starts, and two populations of AChromosomes evolves;
Step 4. According to an individual's AChromosome and its environment, certain activity is to command either Move or Act;
Move: means change to another location;
Act: includes attack, eating, and mating, any one of them three.
Within an individual vision field, no attractive target or food exists, then individual can only choose to Move;
Step 5. Each individual increase *Age* 1; if any one's *Life Span* surpasses *MAX*., then eliminate it from population, also use cadaver as animal food;
Step 6. Every vegetative food increase *Fresh Level* 1; eliminate the expired food supplies which have surpassed their *Time Limit*;
Step 7. Generation number increase 1; if all species extinct or over *MAX* of given generation number, then go to step 3, Loop.

The program iterates to mimic generational evolution over lengthy time frames. Species members experience genetic variations throughout the process, and the survivors remain to reveal which specific genes adapted. Next, we will examine how these remaining genes correlate to produce successful survival.

4. Bayesian modeling of genetic structure

The goal for this experiment is to uncover the hidden relations among AGenes by using BN to analyze the datasets of survivors' AChromosomes. This experiment has an initial run to collect the survivor genes over all generations. As an input, all these AGene expression data, are analyzed by BANANA (BAyesian Network ANAlysis) which incorporates the E-algorithm for BN structure learning. Then a topological structure BN is created to describe the implicit connections among AGenes. ALGAE is a dynamic process based on GA, so the survivors genes will change with each run. We explore the reasons why those survivors prove fittest (best) genes in the AWorld. The underlying principles can be discovered using BN.

4.1. BANANA data processing

To identify the similarities and correlations between the best, fittest genes, is precisely why BN is an appropriate analytical tool. Once a dataset is obtained from ALGAE, it can be analyzed and represented as a Bayesian network. To put the data into usable form, however, requires some manipulation. Firstly, the data containing the genetic information for the AChromosomes must be divided into 12 segments, by bit size, as shown below (Fig. 2):

Then, to facilitate the processing by BANANA, the binary coding of the 12 segments are converted into real values 1 to 4. A conversion principle follows:

if $Seg_i = 00/01$; then $Value_i = 1$;
if $Seg_i = 10/11$; then $Value_i = 2$;
if $Seg_i = 100/101/110/111$; then $Value_i = 3$;

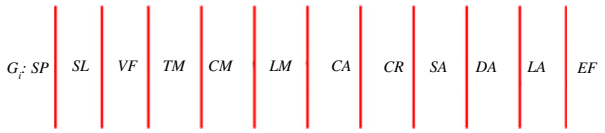
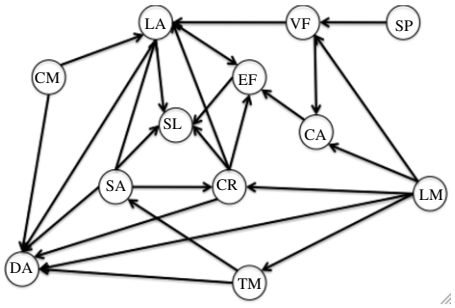


Fig. 2. AChromosome segmented into 12 gene descriptors (see Table 1 for acronym).



SP: SPecies type (1); CA: Action Characteristics(3); SL: Life Span(4); CR: Capricious Rate(6);
VF: Vision Field(4); SA: Attack Speed(5); TM: Transition Movement(3); DA: Defend Ability(6);
CM: Motion Characteristic(2); LA: Attack Loss(7); LM: Motion Loss(5); EF: Food Eciency(4);

Fig. 3. BN model of optimal genetic structure.

Otherwise, $Value_i = 4$.

(Here, Value 1 to 4 is a nominal value.)

Thus, the conversion puts the sequences in 4 different groups: if a sequence equals two bits and the first digit is 0, we set its value as 1; if a sequence equals two bits and the first digit is 1, we set its value as 2; if a sequence equals three digits and starts with 1, we set its values as 3; otherwise, we set a sequence value as 4. We use MS ACCESS database to process the real genotype binary values into integers for the BN analysis.

4.2. BANANA result analysis for gene datasets

BANANA incorporates the E-Algorithm which has been proven to be valid and accurate. First, we collect the data from ALGAE. Ten trials have been undertaken using ALGAE for resolving and analyzing the data. The data is processed by segmenting each chromosome into 12 variables: SP, SL, VF, TM, CM, LM, CA, CR, SA, DA, LA, EF. These 12 variables are also the gene descriptors which originally encoded in the chromosomes from the beginning of AL environment for both species.

BANANA provides a BN representation of 26 arcs connecting the 12 nodes for ALGAE (see Fig. 3). The different numbers show the connecting arcs with other nodes, and we conclude that this reflects the level of effect importance for each gene. Eight arcs connect with defense ability (DA), showing it is the major factor in survival. The second important factor is the energy lost in fighting (LA), with seven arcs. The relationships and dependencies also indicate that speedy attack ability (SA), and the energy cost of survival (EF), with six arcs each, are key genes in determining survival in this chromosome. We therefore see that combat occupies a central role in survival, in a hostile environment with competition for survival. The BN shows the hidden rules of survival which are embedded in the dataset from ALGAE. The rule is that only certain gene combinations will allow a species to survive. That is, defense comes first, and attack skills or energy status affects the "battle period". Every gene has a different level of importance in survival and the evolutionary process, as indicated by the number of connections associated with each gene in Fig. 3. It is the key to why even generations with weaker genes at the outset can adapt to living conditions and live long enough (DA, SL) to have offspring to create the next generation.

The BN structure indicates the formula for the rule of survival: relationship between the characteristics will determine who lives and who does not. Observe that the ability to live long, find food or reproduce (CA) or to defend (DA: protect oneself or ones community) is determined by movement capabilities (CM, TM) and by ability to control energy levels (LM, EF) primarily, and secondarily by how far you can see (VF) and what kind of decisions you make according to individual personality (CR). All these abilities are genetically determined in the AWorld. Chance primarily determines which genes will be present at the start of a run, but it is evolutionary fitness which determines who will actually survive in the world. The BN reveals the hidden structure of fitness. A successful individual's gene composition does not explain the reason for its success. The data merely reveals the principle; however, BN describes the causal relations among the factors and how these connections influence the way the whole diagram works. It shows the reality of why this species could continue to live and thrive. For ALGAE, it is: who adapts and stays to the last, survives!

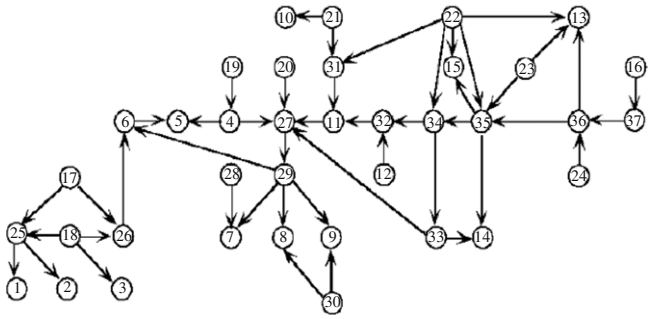


Fig. 4. ALARM network.

5. Conclusion

Bayesian networks in Gene Selection applies BNs to analyze and explain relationships between characteristics of artificial life species. Species can represent any organisms or classes of organism, or any comparable classes of entity existing in a competitive environment. Assuming that evolutionary data is provided, BN analysis assists us to understand the dependencies implicit in the relationships.

First, we provide the E-algorithm for BN structure learning with two noteworthy improvements. One defines a partial structure "Δ-form" for CI tests, in order to reduce redundant causal connections between variables. The second, indicates that the mutual information between each variable and its parents has been ordered and used for a heuristic search to reduce redundant recursions and to solve variable combinatory problems. Experiments on ALARM proves that the E-algorithm is valid, accurate and effective for BN learning.

Furthermore, we implement ALGAE to simulate the viability of two populations in a competitive environment, subject to evolving and adapting forces. ALGAE proves effective at generating data which emulate natural selection and evolution for any two species or entities with definable characteristics. Control of certain factors such as environment, genetic recombination and selection, and presence or absence of specific genes produced valid and reliable data about which genes were fittest, given the constraints of their environment. The dataset favorably compares with standardized datasets.

Thirdly, incorporated with E-algorithm, BANANA is used to analyze the artificial chromosome which is the product of the evolutionary process ALGAE. This research extends the utility of artificial life and the genetic algorithm by capturing and interpreting data which might otherwise be unavailable. This result also provides a unique bridge connecting BN and evolutionary processes. These evolutionary simulation data are useful to researchers who can benefit from predictive modeling.

The experimental results show that Bayesian networks are flexible and valuable analytical data mining tools. The overall results are encouraging and suggest three outcomes: one, a single chromosome or gene combination derived from evolution do not, of themselves, determine fitness or survivability in a given environment. Two, fitness is contingent on the relationship between the AGenes, the mix, and the resulting genotype. Three, BANANA provides a map of the ideal genotype which demonstrates optimal fitness under certain conditions. Thus "optimal" does not mean any particular gene, but a combination of genes.

The process of evolution is accelerated by ALGAE, allowing us to observe generations of genes evolving in a short time. This allows us to foresee the genetic recombination process. We analyze the linkages between generations that favor fitness (and thus survival) which emerge from the data. BN is a critical method to reveal the hidden structure and its relationships, and more importantly, its rules. The principles of how a survivor adapts in evolution from either optimal ancestors or weak ones, and at what point the evolutionary process can be tilted to favor certain adaptive ones, need further research.

Acknowledgements

We would like to thank NSERC and York University for financial and technical support.

Appendix. Experiment results and analysis

We test the E-algorithm in a benchmark ALARM [22] network dataset. ALARM stands for "A Logical Alarm Reduction Mechanism". This is a medical diagnostic system for patient monitoring which contains 37 variables with a set of 2 to 4 values each. Respectively, they are 8 diagnoses, 16 findings and 13 intermediate factors. Fig. 4 is a nontrivial belief network with 46 arcs describing the relationships among these symptoms, the findings and diagnosis for this medical diagnostic system BN representation.

Table 2 explains the details of experiment when E-Algorithm applied to ALARM of 10 000 patient records.

The process starts with creating a complete connection graph of all 37 nodes with 666 arcs. After that, the E-algorithm uses CI constraint test for independent examination among variables based on their mutual information strength. This

Table 2
ALARM test results (threshold $\varepsilon = 0.995$).

Steps	Add arcs	Subtract arcs	Remainder
1. Create complete graph	666	0	666
2. Order-0 independence test	0	373	293
3. Order-1 CI test	0	207	86
4. Order-2 CI test	0	13	73
5. MDL	3	31	45

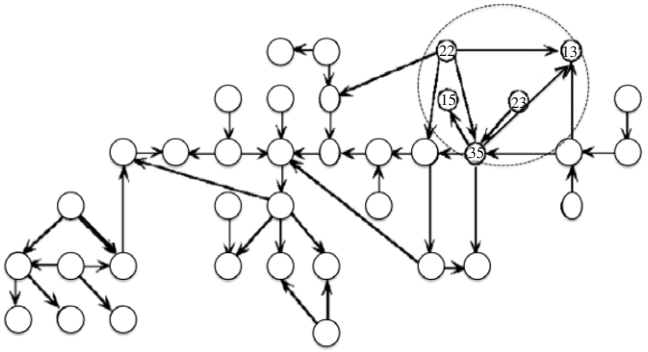


Fig. 5. ALARM network learned by E-algorithm.

removal section separately carries on Order-0 independence test, Order-1 and Order-2 CI test on all arcs, and delete redundant ones, 373, 207 and 13 respectively. The network, after pruned by independent examination, contains 73 arcs. Furthermore, MDL metric function has been applied to evaluate how well the structure fit with the data. It removes 31 arcs and adds 3 more. This whole process obtains a BN structure of 45 arcs (Fig. 5) to represent ALARM.

If we compare BN structure learned by E-algorithm (see Fig. 5) with benchmark ALARM network (Fig. 4), the E-algorithm has created one redundant arc ($35 \rightarrow 13$), and missed two ($22 \rightarrow 15$, $23 \rightarrow 13$). The Fig. 5 structure, literally, does not exactly match the standard in Fig. 4. However, it can be affected by the selected training dataset. The ALARM dataset includes 37 variables; each one has two, three or four possible attributes. Theoretically the possible attribute combination is $2^{13} \times 3^{17} \times 4^7$ possible combinations! We only use a 10 000-record dataset, rather than one of this enormous size and complexity, as it is relatively small. The dataset selection can affect results, since these 10 000 data records may possibly contain a hidden dependence relationship, so may incompletely match the standard ALARM system. Even though our dataset has minor errors, it is well within acceptable ranges.

In experiments on ALARM datasets, the E-algorithm has proved that it is efficient, valid, and produces high accuracy for learning BN.

References

[1] R. Lewontin, The Genetic Basis of Evolutionary Change, Columbia University Press, 1974.
[2] R.E. Michod, Positive heuristics in evolutionary biology, The British Journal for the Philosophy of Science 32 (1) (1981) 1–36.
[3] L. Partridge, P.H. Harvey, Evolutionary biology: Costs of reproduction, Nature 316 (1985) 20–21.
[4] C.M. Bishop, Pattern Recognition and Machine Learning (Information Science and Statistics), Springer, 2006.
[5] G.F. Cooper, E. Herskovits, A Bayesian method for constructing Bayesian belief networks from databases, in: G.F. Cooper, S. Moral (Eds.), Proceedings of the Seventh Conference (1991) on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1991, pp. 86–94.
[6] G.F. Cooper, E. Herskovits, A bayesian method for the induction of probabilistic networks from data, Machine Learning 09 (4) (1992) 309–347. URL: http://www.ingentaconnect.com/content/klu/mach/1992/00000009/00000004/00422779.
[7] L.M. de Campos, J.M. Fernández-Luna, J.F. Huete, Bayesian networks and information retrieval: An introduction to the special issue, Information Processing & Management 40 (5) (2004) 727–733.
[8] N. Friedman, I. Nachman, D. Pe'er, Learning Bayesian Network Structure from Massive Datasets: The "Sparse Candidate" Algorithm, 1999, pp. 206–215. URL: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.37.5605.
[9] D. Heckerman, D. Geiger, D.M. Chickering, Learning bayesian networks: The combination of knowledge and statistical data, Machine Learning 20 (3) (1995) 197–243. URL: http://www.ingentaconnect.com/content/klu/mach/1995/00000020/00000003/00422402.
[10] J. Ji, C. Liu, J. Yan, N. Zhong, Bayesian networks structure learning and its application to personalized recommendation in a b2c portal, in: WI '04: Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence, IEEE Computer Society, Washington, DC, USA, 2004, pp. 179–184.
[11] M. Pelikan, Probabilistic model-building genetic algorithms, in: GECCO '08: Proceedings of the 2008 GECCO Conference Companion on Genetic and Evolutionary Computation, ACM, New York, NY, USA, 2008, pp. 2389–2416.
[12] M. Mitchell, S. Forrest, Genetic algorithms and artificial life, Artificial Life 1 (3) (1994) 267–289.
[13] J. Yan, Bayesian Network Structure Learning, Bachelor Thesis, Beijing University of Technology, 2003.
[14] J. Yan, S. Lv, N. Zhong, Artificial life modeling in corporate strategy, Journal of Guangxi Normal University 25 (4) (2007).
[15] J. Pearl, A constraint-propagation approach to probabilistic reasoning, in: L.N. Kanal, J.F. Lemmer (Eds.), Uncertainty in Artificial Intelligence, North-Holland, Amsterdam, 1986, pp. 357–369.
[16] D. Heckerman, A tutorial on learning with bayesian networks, Tech. rep., Microsoft Research, Redmond, Washington, 1995. URL: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.56.1431.

[17] J. Cheng, D.A. Bell, W. Liu, Learning belief networks from data: An information theory based approach, in: Proceedings of the Sixth ACM International Conference on Information and Knowledge Management, 1997, pp. 325–331.
[18] J. Cheng, R. Greiner, J. Kelly, D. Bell, W. Liu, Learning bayesian networks from data: An information-theory based approach, Artificial Intelligence 137 (1–2) (2002) 43–90.
[19] L. Qiang, T.Y. Xiao, G.X. Qiao, An improved bayesian networks learning algorithm, Journal of Computer Research and Development 39 (10) (2002) 1221–1226.
[20] J. Suzuki, Learning bayesian belief networks based on the minimum description length principle: Basic properties, IEICE Transactions on Fundamentals E82 10 (1999) 2237–2245.
[21] M.L. Wong, W. Lam, K.S. Leung, Using evolutionary programming and minimum description length principle for data mining of bayesian networks, IEEE Transactions Pattern Analysis and Machine Intelligence 21 (2) (1999) 174–178.
[22] I.A. Beinlich, H.J. Suermondt, R.M. Chavez, G.F. Cooper, The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks, in: Second European Conference on Artificial Intelligence in Medicine, vol. 38, Springer-Verlag, Berlin, London, Great Britain, 1989, pp. 247–256.
[23] J. Ji, J. Yan, C. Liu, N. Zhong, An improved bayesian networks learning algorithm based on independence test and mdl scoring, in: Proceedings of the 2005 International Conference on Active Media Technology, AMT 2005, 2005, pp. 315–320.
[24] N. Friedman, M. Linial, I. Nachman, D. Pe'er, Using bayesian networks to analyze expression data, Journal of Computational Biology 7 (2000) 601–620.
[25] G. Liu, W. Feng, H. Wang, L. Liu, C. Zhou, Reconstruction of gene regulatory networks based on two-stage bayesian network structure learning algorithm, Journal of Bionic Engineering 6 (1) (2009) 86–92. URL: http://www.sciencedirect.com/science/article/B82XN-4W26FGG-F/2/23d9d555770.
[26] M. Wang, Z. Chen, S. Cloutier, A hybrid bayesian network learning method for constructing gene networks, Computational Biology and Chemistry 31 (5–6) (2007) 361–372.